

SAT je NP-těžký

$SAT = \{\langle \varphi \rangle \mid \varphi \text{ je splnitelná Booleovská formule}\}$

SAT je NP-těžký, tj. $A \in NP \implies A \leq_p SAT$:

Nechť \mathcal{N} je nedeterministický TM rozhodující A v čase n^k . Pro každé slovo w sestrojíme Booleovskou formuli Φ , která je splnitelná, právě když stroj \mathcal{N} má akceptující výpočet nad w .

SAT je NP-těžký

Každý výpočet stroje \mathcal{N} pracujícího v čase n^k nad slovem w lze reprezentovat tablem:

Vytvoříme Φ , aby platilo:

Φ je splnitelné \iff existuje akceptující tablo reprezentující výpočet na w

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

Φ_{cell} = “každé $x_{i,j,s}$ platí \iff v tablu na pozici i, j je symbol s ,
kde $s \in C = Q \cup \Gamma \cup \{\#\}$ ”

$$\Phi_{\text{cell}} = \bigwedge_{\substack{1 \leq i \leq n^k + 1 \\ 1 \leq j \leq n^k + 4}} \left[\left(\bigvee_{s \in C} x_{i,j,s} \right) \wedge \left(\bigwedge_{\substack{s, t \in C \\ s \neq t}} \neg(x_{i,j,s} \wedge x_{i,j,t}) \right) \right]$$

$$|\Phi_{\text{cell}}| \in \mathcal{O}(n^{2k})$$

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

Φ_{start} = “na prvním řádku je iniciální konfigurace pro $w = w_1 w_2 \dots w_n$ ”

$$\begin{aligned} \Phi_{\text{start}} = & X_{1,1,\#} \wedge X_{1,2,q_0} \wedge \\ & X_{1,3,w_1} \wedge X_{1,4,w_2} \wedge \dots \wedge X_{1,n+2,w_n} \wedge \\ & X_{1,n+3,\sqcup} \wedge X_{1,n+4,\sqcup} \wedge \dots \wedge X_{1,n^k+3,\sqcup} \wedge X_{1,n^k+4,\#} \end{aligned}$$

$$|\Phi_{\text{start}}| \in \mathcal{O}(n^k)$$

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

Φ_{move} = “každé dva po sobě jdoucí řádky tabla odpovídají kroku výpočtu”

popíšeme pomocí “legálních oken” 2×3

příklady legálních oken pro $\delta(q_1, b) = \{(q_2, c, L), (q_2, a, R)\}$:

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

Φ_{move} = “každé okno tabla je legální” (okna se překrývají)

$$\Phi_{\text{move}} = \bigwedge_{\substack{1 \leq i < n^k + 1 \\ 1 < j < n^k + 4}} \left(\bigvee_{\text{legální okno}} \begin{array}{c} X_{i,j-1,a_1} \wedge X_{i,j,a_2} \wedge X_{i,j+1,a_3} \\ X_{i+1,j-1,a_4} \wedge X_{i+1,j,a_5} \wedge X_{i+1,j+1,a_6} \end{array} \right)$$

a_1	a_2	a_3
a_4	a_5	a_6

$$|\Phi_{\text{move}}| \in \mathcal{O}(n^{2k})$$

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

$\Phi_{\text{accept}} =$ “v tablu je stav q_{acc} ”

$$\Phi_{\text{accept}} = \bigwedge_{\substack{1 < i \leq n^k + 1 \\ 1 < j < n^k + 4}} x_{i,j,q_{\text{acc}}}$$

$$|\Phi_{\text{accept}}| \in \mathcal{O}(n^{2k})$$

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

$$|\Phi| = \mathcal{O}(n^{2k}) + \mathcal{O}(n^k) + \mathcal{O}(n^{2k}) + \mathcal{O}(n^{2k}) = \mathcal{O}(n^{2k})$$

Počet proměnných $x_{i,j,s}$ závisí na $n = |w|$ a není pevně omezen.
Proměnnou lze zakódovat $\mathcal{O}(\log n)$ znaky.

Tedy $|\langle \Phi \rangle| = \mathcal{O}(n^{2k} \log n)$.

$\langle \Phi \rangle$ má polynomiální délku vzhledem k $n = |w|$ a lze spočítat v polynomiálním čase. Tedy $A \leq_p SAT$.

Konjunktivní normální forma (cnf) formulí

literál = je proměnná nebo její negace

klauzule = disjunkce literálů

formule v cnf = konjunkce klauzulí

formule v 3cnf = formule v cnf, kde všechny klauzule obsahují 3 literály

Problém 3SAT

Problém 3SAT je problém rozhodnout, zda je daná Booleovská formule v 3cnf formě splnitelná.

$$3SAT = \{ \langle \varphi \rangle \mid \varphi \text{ je splnitelná formule v 3cnf} \}$$

Věta. 3SAT je NP-úplný.

Důkaz. 3SAT \in NP:

3SAT je NP-těžký

$3SAT = \{\langle \varphi \rangle \mid \varphi \text{ je splnitelná formule v 3cnf}\}$

3SAT je NP-těžký, tj. $A \in NP \implies A \leq_p 3SAT$:

1 zkonstruujeme Φ jako v důkazu NP-těžkosti SATu

2 Φ převedeme na ekvivalentní Φ' v cnf pomocí ekvivalence

$$(\varphi \wedge \psi) \vee \rho \equiv (\varphi \vee \rho) \wedge (\psi \vee \rho)$$

3SAT je NP-těžký

3 Φ' převedeme na ekvivalentní Φ'' v 3cnf pomocí ekvivalencí

$$\begin{array}{c} l_1 \\ l_1 \vee l_2 \\ l_1 \vee l_2 \vee l_3 \\ l_1 \vee l_2 \vee l_3 \vee l_4 \\ \vdots \\ l_1 \vee l_2 \vee \dots \vee l_m \end{array}$$

Φ'' je ekvivalentní Φ a $|\Phi''| \in \mathcal{O}(n^{2k})$. Tedy $A \leq_p 3SAT$. □

Redukce a \mathcal{C} -těžkost

Lemma. Necht' \mathcal{C} je složitostní třída.

A je \mathcal{C} -těžký a $A \leq_p B \implies B$ je \mathcal{C} -těžký

Důkaz.



Důsledek. Je-li A NP-úplný, $A \leq_p B$ a $B \in \text{NP}$, pak B je také NP-úplný.

Prostorová složitost algoritmu

- paměť použitá při výpočtu
- závisí na vstupu
- jako základní model použijeme **Turingův stroj**
- zkoumáme **nejhorší případ**, tedy maximální počet přečtených políček pásky v závislosti na délce vstupu
- lze zkoumat i **průměrný případ**

Prostorová složitost Turingova stroje

Definice. Necht' \mathcal{M} je úplný deterministický (jednopáskový nebo vícepáskový) Turingův stroj se vstupní abecedou Σ . Pro každé $w \in \Sigma^*$ definujeme $s_{\mathcal{M}}(w)$ jako počet políček pásky, které stroj \mathcal{M} čte při výpočtu na vstupu w . **Prostorová složitost** stroje \mathcal{M} je pak funkce $S_{\mathcal{M}} : \mathbb{N}_0 \rightarrow \mathbb{N}$ definovaná vztahem

$$S_{\mathcal{M}}(n) = \max\{s_{\mathcal{M}}(w) \mid w \in \Sigma^n\}.$$

Definice. Definice **prostorové složitosti** úplného nedeterministického Turingova stroje se liší pouze tím, že $s_{\mathcal{M}}(w)$ označuje maximální počet políček pásky, které stroj \mathcal{M} čte při nějakém výpočtu na vstupu w .

Používáme asymptotickou notaci, protože prostor lze komprimovat.

Příklad

$$\mathcal{M} = (\{q_0, q_1, q_{\text{acc}}, q_{\text{rej}}\}, \{0, 1\}, \{0, 1, \triangleright, \sqcup\}, \triangleright, \sqcup, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$$

δ	\triangleright	0	1	\sqcup
q_0	(q_0, \triangleright, R)	$(q_0, 0, R)$	$(q_1, 1, R)$	$(q_{\text{acc}}, -, -)$
q_1		$(q_{\text{rej}}, -, -)$	$(q_1, 1, R)$	$(q_{\text{acc}}, -, -)$

Kompresce prostoru

Věta. Pro každý deterministický úplný TM \mathcal{M} a pro každé $m > 1$ lze zkonstruovat deterministický úplný TM \mathcal{M}' tak, že $L(\mathcal{M}) = L(\mathcal{M}')$ a

$$S_{\mathcal{M}'}(n) = \frac{S_{\mathcal{M}}(n)}{m} + n + 2.$$

Důkaz.



Prostorové složitostní třídy problémů

prostorová složitost problému = nejmenší prostorová složitost, s jakou lze daný problém rozhodnout

Definice. Každá funkce $f : \mathbb{N} \rightarrow \mathbb{N}$ definuje **prostorové složitostní třídy problémů**:

$\text{SPACE}(f(n)) = \{L \mid L \text{ je rozhodovaný nějakým deterministickým TM } \mathcal{M} \text{ s prostorovou složitostí } S_{\mathcal{M}}(n) = \mathcal{O}(f(n))\}$

$\text{NSPACE}(f(n)) = \{L \mid L \text{ je rozhodovaný nějakým nedet. TM } \mathcal{N} \text{ s prostorovou složitostí } S_{\mathcal{N}}(n) = \mathcal{O}(f(n))\}$

$SAT \in \mathbf{SPACE}(n)$

$SAT = \{ \langle \varphi \rangle \mid \varphi \text{ je splnitelná Booleovská formule} \}$

SAT může být rozhodován deterministickým třípáskovým TM \mathcal{M} :

- na 2. pásku postupně zapisujeme všechna ohodnocení proměnných
- na 3. pásce pro dané ohodnocení ověříme, zda splňuje vstupní formuli
- akceptujeme, pokud narazíme na splňující ohodnocení
- zamítneme, pokud žádné ohodnocení není splňující

Prostor lze použít opakovaně.

Čas a prostor

Věta. Pro každou funkci $f : \mathbb{N} \rightarrow \mathbb{N}$ platí:

- 1 $\text{TIME}(f(n)) \subseteq \text{SPACE}(f(n))$
- 2 $\text{NTIME}(f(n)) \subseteq \text{NSPACE}(f(n))$
- 3 $\text{SPACE}(f(n)) \subseteq \text{TIME}(k^{f(n)})$ pro vhodné $k \in \mathbb{N}$
- 4 $\text{NSPACE}(f(n)) \subseteq \text{NTIME}(k^{f(n)})$ pro vhodné $k \in \mathbb{N}$

Důkaz.



Savitchova věta

Savitchova věta. Pro každou funkci $f : \mathbb{N} \rightarrow \mathbb{N}$ splňující $f(n) \geq n$ platí:

$$\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$$

Standardní převod nedeterministického TM na deterministický nefunguje:

Savitchova věta

Důkaz. Necht' \mathcal{N} je nedeterministický TM s prostorovou složitostí $f(n)$. Stroj upravíme tak, aby před akceptováním smazal pásku a posunul hlavu zcela vlevo. Má tedy jen jednu akceptující konfiguraci c_{acc} . Výpočet stroje má maximálně $k^{f(n)}$ kroků.

Ekvivalentní deterministický stroj \mathcal{M} implementuje proceduru $\text{comp}(c_1, c_2, t)$, která akceptuje, pokud lze ve stroji \mathcal{N} během t kroků přejít z konfigurace c_1 do c_2 , jinak zamítá. Je-li c_w iniciální konfigurace stroje \mathcal{N} pro w , stačí tedy spustit $\text{comp}(c_w, c_{\text{acc}}, k^{f(n)})$.

$\text{comp}(c_1, c_2, t)$ lze implementovat rekursivně:

Savitchova věta

Algoritmus pro $comp(c_1, c_2, t)$:

- 1 $t = 1$: Otestujeme, zda platí $c_1 = c_2$ nebo $c_1 \not\equiv_{\mathcal{N}} c_2$.
Pokud platí, akceptujeme, jinak zamítneme.
- 2 $t > 1$: Pro každou konfiguraci c' stroje \mathcal{N} využívající max. $f(n)$ políček
- 3 spustíme $comp(c_1, c', \lceil \frac{t}{2} \rceil)$ a $comp(c', c_2, \lceil \frac{t}{2} \rceil)$.
- 4 Pokud obojí akceptuje, akceptujeme.
- 5 Pokud žádné c' nevedlo k akceptování, zamítneme.

Prostorová složitost:

- $comp$ potřebuje prostor na c_1, c_2, c' a t (a něco konstantního):
- hloubka rekurzivního volání:
- celkem:



Polynomiální prostorové složitostní třídy

$$\text{PSPACE} = \bigcup_{k \in \mathbb{N}} \text{SPACE}(n^k)$$

$$\text{NPSPACE} = \bigcup_{k \in \mathbb{N}} \text{NSPACE}(n^k)$$

Věta. $\text{PSPACE} = \text{NPSPACE}$

Důkaz. Plyne přímo ze Savitchovy věty.



Vztahy prostorových a časových tříd

$P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME \subseteq NEXPTIME$

Problém TQBF

QBF = kvantifikované Booleovské formule (proměnné v doméně $\{0, 1\}$)

$$\exists x \forall y ((x \vee y) \wedge (\neg x \vee \neg y))$$

Definice. Problém TQBF je problém rozhodnout, zda je daná QBF formule bez volných proměnných pravdivá.

$$TQBF = \{ \langle \varphi \rangle \mid \varphi \text{ je pravdivá QBF formule bez volných proměnných} \}$$

Věta. $TQBF$ je PSPACE-úplný.

Důkaz. Ukážeme, že $TQBF \in \text{PSPACE}$ a že $TQBF$ je PSPACE-těžký. \square

$TQBF \in PSPACE$

$TQBF$ lze rozhodovat pomocí rekurzivní procedury $t(\varphi)$:

- 1 Pokud φ neobsahuje kvantifikátory, snadno vyhodnotíme a akceptujeme, pokud je φ pravdivá. Jinak zamítneme.
- 2 Je-li φ tvaru $\exists x(\varphi')$, spustíme $t(\varphi'[x \mapsto 0])$ a $t(\varphi'[x \mapsto 1])$. Pokud alespoň jedno z volání akceptuje, akceptujeme. Jinak zamítneme.
- 3 Je-li φ tvaru $\forall x(\varphi')$, spustíme $t(\varphi'[x \mapsto 0])$ a $t(\varphi'[x \mapsto 1])$. Pokud obě volání akceptují, akceptujeme. Jinak zamítneme.

Prostorová složitost:

- v jednom zavolání t si stačí pamatovat pouze něco konstantního.
- hloubka rekurzivního volání:
- celkem:

TQBF je PSPACE-těžký

Ukážeme $A \in \text{PSPACE} \implies A \leq_p \text{TQBF}$.

Nechť \mathcal{M} je TM s prostorovou složitostí n^k rozhodující A . Tedy \mathcal{M} pracuje v čase $d^{(n^k)}$. Předpokládáme, že \mathcal{M} má jednu akceptující konfiguraci c_{acc} . Důkaz kombinuje myšlenky důkazů NP-těžkosti SATu a Savitchovy věty.

Pro každé slovo w sestrojíme QBF formuli $\Phi_{c_{\text{start}}, c_{\text{acc}}, d^{(n^k)}}$, která je pravdivá, právě když existuje výpočet stroje \mathcal{M} z iniciální konfigurace pro w c_{start} do nějaké akceptující konfigurace s nejvýše $d^{(n^k)}$ kroky.

TQBF je PSPACE-těžký

Formuli budujeme induktivně:

■ $\Phi_{c_1, c_2, 1}$ je pravdivá právě když $c_1 = c_2$ nebo $c_1 \perp_{\mathcal{M}} c_2$.

■ pro $t > 1$: $\Phi_{c_1, c_2, t} =$

$$\exists m \forall (c_3, c_4) \in \{(c_1, m), (m, c_2)\} \left(\Phi_{c_3, c_4, \lceil \frac{t}{2} \rceil} \right)$$

$|\Phi| =$

Tedy $|\Phi|$ je polynomiální vzhledem k $|w| = n$ a je pravdivá $\iff w \in A$.
Celkem $A \leq_p TQBF$ pro každé $A \in PSPACE$.